

REMARKS

Claims 2-8 are pending in the present application. The Office Action and cited references have been considered. Favorable reconsideration is respectfully requested.

Claims 2-4 and 8 were rejected under 35 U.S.C. §102(b) as being anticipated by Cirne (U.S. Patent No. 6,260,187 - hereinafter "Cirne"). Claims 5 and 6 were rejected under 35 U.S.C. §103(a) as being unpatentable over Cirne in view of official notice. These rejections are respectfully traversed for the following reasons.

Before proceeding to the merits, the first paragraph of section 3 of the office action refers to claims 2-4 and 8. However, the body of the rejection refers also to claim 7. Applicant assumes that the first paragraph should have also referred to claim 7.

In pertinent part, claim 7 is directed to a method for loading an updated release of an earlier application into a computer device, said earlier application having earlier application classes and earlier static field identifiers and said updated release having updated application classes and updated static field identifiers. Claim 7 has been modified to clarify the fact that the application was previously loaded in the computer device, not only in the sense that its code was present in some storage medium or memory of the device, but in the stronger sense that the application had previously been executed on the computer device and that data (in the form of objects) of the earlier application release is present in some storage medium or memory of the computer device. For example, in the case of a chip card application as described by Applicant, it is essential that an update to the application conserves or adapts the data stored by the earlier application release, rather than deleting and reloading the data (¶0005 of the published application, no. 2007/0277168). Loading the updated application on the computer device

comprises not only copying the code into a storage medium or memory of this computer device, but also initializing static fields of the updated classes (using already stored values for inherited fields) (§0063) and modifying existing objects to make use of the updated classes (§0064).

The method recited in claim 7 comprises four steps, here numbered for easy reference:

1. computing in a first computing operation prior to said loading a class matching information establishing a correspondence between said earlier application release classes and said updated application release classes.
2. computing in a second computing operation, prior to said loading a second static field identifiers matching information establishing a correspondence between said earlier application release static field identifiers and said updated application release static field identifiers.
3. linking said class matching information and said static field identifiers matching information to said updated application release as loaded into the computer device.
4. using said class matching information and said static field identifiers matching information to modify said objects to point at the updated application release classes and use the updated application release static field identifiers.

Steps 1 and 2 recite the computing of correspondences between characteristics of the earlier application release and the updated application release.

Step 3 recites the linking of the correspondences computed in steps 1–2 “to said updated application release as loaded into the computer device”.

Step 4 recites the use of the correspondences computed in steps 1–2 to modify objects of the earlier application release so that they become objects of the updated application release. That is to say, objects already residing on the computer device are modified to use the updated application classes (published application, ¶0064).

Cirne teaches a method for “modifying object oriented code,” comprising “substitut[ing] the allocation of an object of a new class for the allocation of the object based on an original class,” “chang[ing] code that allocates an object of an original class to code that calls a static method in another class that allocates the object of the original class,” and “replac[ing] a new static field for an original static field.” This method only involves modifications of executable code (Cime, col. 2, ll. 50–67; Cime, fig. 6–8). In contrast, Applicant’s invention involves modifications of data stored in the memory of a computer device.

Claim rejections — 35 USC §102(b)

As per claim 7, Cirne does not teach that data may be modified as part of an application update. Indeed, Cirne’s transformation operates on application code alone, without said code having previously been run to produce or update data in the form of objects of classes of the original application or static fields in classes of the original application. Therefore, step 4 is not disclosed by Cirne. Thus, Cirne does not disclose claim 7. Moreover, Cirne does not even suggest that data may be modified as part of an application update. Thus, step 4 is not even suggested by Cirne.

As per claim 2-4 and 8, claims 2-4 and 8 depend from and include the limitations of claim 1. Thus, claims 2-4 and 8 are believed to be patentable in and of themselves and as they depend from and include the limitations of claim 7.

Claim rejections — 35 USC §103(a)

As Cirne does not even suggest that data may be modified as part of an application update, step 4 of claim 7 is not even suggested by Cirne, and the subject matter of claim 7 would not have been obvious to one having ordinary skill in the art. Claims 5 and 6 depend from and include the limitations of claim 1. Thus, Cirne does not disclose nor suggest subject matter of claim 5 or 6 and the subject matter of claims 5 and 6 would not have been obvious to one having ordinary skill in the art.

Cirne teaches a method for transforming program code before it is run. In particular, Cirne does not teach running or even loading both the original program code and the updated program code on the same computer device. On the other hand, Applicant's method concerns loading and executing an updated application on a computer device on which the original application was previously loaded and, preferably, run (§0011).

Cirne does not teach nor suggest a method for loading an updated version of an original one, while this original application runs.

Cirne only teaches a transformation of program code, whereas Applicant's invention deals with a transformation of program data. Applicant's invention comprises a step of modifying object data stored on the computer device on which the original program was loaded in order for the resulting data to be suitable for the updated application (§0011; claim 7). This point is very relevant in practice because chip card applications may store confidential data (such

as secret cryptographic keys) that never leaves the chip card after it has been produced under highly secure conditions (§0005).

Cirne does not teach nor suggest a method for loading an updated version of an original one, without uploading and/or downloading confidential data already stored on the chip card. Applicant respectfully submits that the purpose of the method according to Cirne is quite different from that of Applicant's method. For this reason, the teaching of Cirne would not have been used by one of ordinary skill in the art to solve the problems faced by Applicant.

In view of the above amendment and remarks, Applicant respectfully requests reconsideration and withdrawal of the outstanding rejections of record. Applicant submits that the application is in condition for allowance and early notice to this effect is most earnestly solicited.

If the Examiner has any questions, he is invited to contact the undersigned at 202-628-5197.

Respectfully submitted,

BROWDY AND NEIMARK, P.L.L.C.
Attorneys for Applicant(s)

By /Ronni S. Jillions/
Ronni S. Jillions
Registration No. 31,979

RSJ:ma
Telephone No.: (202) 628-5197
Facsimile No.: (202) 737-3528
G:\B\NM\Mout\Veillard\1\Pro\2009-05-18-Amendment\VEILLARD1.doc